

Regression Analysis Application Using ML.NET

Prof. Khot Jayashri S.¹, Dr. Divekar Sudhir N.², Hande Sagar R.³, Holkar Soham V.⁴, Inamdar Sufiyan S.⁵

Department of Electronics and Computer Engineering , HSBPVT's GOI, Faculty of Engineering , Kashti

Abstract— The rapid proliferation of machine learning (ML) technologies has created significant demand for accessible, flexible, and intelligent data analysis platforms. This research presents an innovative web-based Dual-Approach Analytics Application that uniquely combines both AI-automated and manual analysis workflows. Developed using ML.NET, React, and a Claude AI-driven analytics engine, the system offers users two distinct pathways for data analysis: (1) **AI-Powered Approach** - where AI automatically identifies target columns, features, and encodings, constructing ML pipelines autonomously; and (2) **Manual Approach** - providing data scientists and analytics professionals with granular control over column selection, encoding strategies, algorithm choice, and visualization options. Users authenticate securely using JWT-based access and refresh tokens, upload CSV datasets, and choose their preferred analysis pathway. The AI-powered approach achieves prediction accuracy exceeding 93-95% with zero configuration, while the manual approach enables expert users to leverage domain knowledge for optimized model performance. Both approaches deliver results through rich, interactive dashboard visualizations built with React. This dual-pathway architecture democratizes ML for novices while empowering experts with professional-grade analytics tools in a unified platform. Experimental evaluations across multiple real-world datasets confirm the system's robustness, flexibility, accuracy, and comprehensive applicability across diverse user expertise levels.

Key Words: : ML.NET, Dual-Approach Analytics, AI-Powered Analysis, Manual ML Configuration, CSV Analytics, JWT Authentication, React, Dynamic Machine Learning, Interactive Data Analysis, Professional Analytics Tools, Automated and Manual Workflows.

1. INTRODUCTION

The convergence of artificial intelligence, machine learning, and modern web technologies has opened unprecedented opportunities for automating complex analytical tasks while simultaneously providing sophisticated tools for expert users. Traditional data analytics platforms typically serve one extreme: either they are overly simplified “black box” solutions that limit expert control, or they are complex frameworks requiring extensive programming knowledge. This research presents a novel **Dual-Approach Analytics Platform** that bridges this divide, offering both fully automated AI-driven analysis and comprehensive manual configuration within a single, unified system.

Machine learning with .NET has gained substantial traction since Microsoft introduced ML.NET, a cross-platform, open-source ML framework for .NET developers. ML.NET enables developers to build, train, and deploy ML models directly within the .NET ecosystem without requiring Python or external ML libraries. Combined with the AI analytical capabilities of large language models and the dynamic visualization power of React, ML.NET forms the backbone of a comprehensive, production-ready dual-approach analytics application.

The Dual-Approach Innovation

The system presented in this paper introduces a paradigm shift by offering users **two distinct analysis pathways**:

AI-Powered Approach (Automated Path)

In this mode, users simply upload a CSV file and the AI engine handles all analytical decisions. The Claude AI component intelligently identifies the appropriate target (label) column, determines which columns serve as predictive features, recommends categorical encoding strategies for non-numeric columns, suggests optimal algorithms, and even selects appropriate visualization types. The ML.NET pipeline is constructed dynamically based on these AI recommendations, trained on the uploaded data, and evaluated for prediction accuracy—all without any user configuration. This approach democratizes ML for business analysts, domain experts, and non-technical users.

Manual Approach (Expert Path)

For data scientists, ML engineers, and analytics professionals, the system provides comprehensive manual controls. Users can: - **Manually select the target column** from a visual interface displaying all column statistics (data types, null counts, unique values, ranges) - **Choose specific encoding strategies** for each column from 15+ options including Label Encoding, One-Hot Encoding, Binary Encoding, Target Encoding, Frequency Encoding, and advanced techniques like WOE (Weight of Evidence) - **Select from 20+ ML algorithms** spanning regression (Linear, Ridge, Lasso, Polynomial), classification (Logistic, Decision Tree, Random Forest, SVM, XGBoost, LightGBM), and clustering (K-Means, DBSCAN, Hierarchical) - **Configure algorithm hyperparameters** for model optimization - **Select visualization types** from 50+ chart options including comparison charts, trend analysis, distribution plots, composition charts, and advanced analytics visualizations

This manual pathway empowers experts to apply domain knowledge, test hypotheses, and fine-tune models beyond what AI automation can achieve.

Both approaches deliver results through the same rich, interactive React dashboard, ensuring a consistent user experience regardless of the chosen pathway. The dual-approach architecture recognizes that different users have different needs: novices require simplicity, while experts demand control—and the same platform serves both seamlessly

1.1 Motivation

Despite the abundance of ML frameworks, most existing solutions force users into a single paradigm. AutoML platforms like Google AutoML and Azure ML Studio provide automation but severely limit expert control over preprocessing and algorithm selection. Conversely, traditional ML frameworks like scikit-learn and PyTorch demand extensive coding expertise, making them inaccessible to non-technical professionals. No existing platform successfully combines both approaches.

Non-technical professionals in domains such as healthcare, finance, education, and retail often have access to valuable datasets but lack the expertise to extract predictive insights through code-based ML workflows. Simultaneously, data science teams require flexible, professional-grade tools that accelerate their workflows without restricting their ability to apply domain knowledge and advanced techniques. This dual-approach system is the first platform to serve both audiences within a unified architecture.

1.2 Objectives

The primary objectives of this research are:

- To design and implement a **dual-pathway analytics platform** offering both AI-automated and manual analysis modes within a unified full-stack application using ML.NET for the backend and React for the frontend.
- To implement a secure authentication system using JWT with both access tokens and refresh tokens, ensuring session continuity and security across both analysis pathways.
- To integrate an **AI-based automated analysis engine** that dynamically identifies target columns, features, encoding requirements, algorithm selection, and visualization recommendations for zero-configuration analytics.
- To develop a **comprehensive manual configuration interface** providing data professionals with granular control over column analysis, 15+ encoding strategies, 20+ ML algorithms with hyperparameters, and 50+ visualization options.
- To achieve prediction accuracy exceeding 93% in automated mode while enabling expert optimization to reach 95-98% accuracy through manual refinement in the professional pathway.
- To create an **intelligent UI that adapts** based on user selection, providing simplified workflows for AI-powered mode and detailed configuration screens for manual mode.
- To visualize results through interactive, customizable React dashboards that serve both novice and expert users effectively, replacing traditional analytics reports

2. LITERATURE REVIEW

Automated machine learning (AutoML) and manual ML configuration represent two historically separate research domains. Feurer et al. [1] introduced Auto-sklearn, which automates algorithm selection and hyperparameter tuning but operates exclusively within Python and provides no mechanism for manual override or expert configuration. The

ML.NET framework by Microsoft [2] demonstrated that performant ML can be integrated directly into .NET applications but lacks automated feature engineering and requires manual pipeline construction.

Google’s Cloud AutoML [3] pioneered commercial AutoML solutions, automating model training for vision, natural language, and structured data—but restricts users to predefined workflows with no expert override capabilities. Azure Machine Learning Studio [4] provides drag-and-drop automation for non-coders and code-based notebooks for experts, but these remain separate interfaces rather than an integrated dual-approach system.

Hybrid approaches have been explored. Chen et al. [5] proposed “Human-in-the-Loop AutoML” allowing users to guide automated processes, but their system focuses on algorithm selection rather than providing full manual configuration options. Zöller and Huber [6] surveyed AutoML benchmarks and noted that no existing system successfully combines full automation with comprehensive manual controls, identifying this as a critical research gap.

JWT-based authentication has become the de facto standard for securing stateless web APIs. Jones et al. [7] formalized the JSON Web Token specification (RFC 7519), which underpins the dual-token (access and refresh) strategy employed in this application for secure and continuous user sessions. Research by Wieringa et al. [8] confirms that refresh token rotation significantly reduces security risks associated with long-lived sessions.

The use of large language models (LLMs) for data analysis tasks has grown rapidly. Narayan et al. [9] explored LLM-based data cleaning and schema inference, demonstrating that AI models can accurately interpret tabular data structure and suggest preprocessing strategies. This informed the AI analysis engine in our automated pathway. For visualization, Satyanarayan et al. [10] established best practices for interactive data visualization in web-based environments, which guided both the automated and

manual dashboard designs.

Sr. No.	Title & Author	Study Focus	Key Findings	Limitations	Addressed by Our System
1	Auto-sklearn: Efficient AutoML — Feurer et al. (2015)	Automated algorithm selection and optimization	Reduces human effort systematically	Python-only; no manual override for experts	Dual-approach: AI + full manual control
2	ML.NET: ML for .NET — Microsoft (2023)	Cross-platform ML framework	Native .NET integration	No AutoML; requires manual pipeline design	Automated AI path + manual path with ML.NET
3	Google Cloud AutoML — Google (2018)	Commercial AutoML platform	Accessible ML for non-coders	Zero manual control for experts; black-box	AI automation + expert configuration
4	Azure ML Studio — Microsoft (2023)	Hybrid ML platform	Separate interfaces for GUI and code	Disjointed experiences; not unified	Single unified platform with two modes
5	Human-in-the-Loop AutoML — Chen et al. (2021)	Semi-automated ML with guidance	Users can guide automation	Limited to algorithm selection only	Full manual control: columns, encodings, algorithms, visualizations

6	LLM-based Data Inference — Narayan et al. (2022)	AI for tabular data understanding	LLMs infer column semantics accurately	Computational overhead	Optimized AI analysis for CSV datasets
---	--	-----------------------------------	--	------------------------	--

Table 1 — Literature Review

3. PROPOSED SYSTEM ARCHITECTURE

The Dual-Approach Analytics Application adopts a sophisticated three-tier architecture with a unique bifurcation layer that routes users through either the AI-automated or manual analysis pathway based on their selection. The architecture separates the presentation layer (React frontend with dual interfaces), the application and ML processing layer (ASP.NET Core + ML.NET backend with dual pipelines), and the data persistence layer (database + file storage).

Frontend — React with Dual Interface Modes

The React frontend provides two distinct user interface flows within a unified single-page application (SPA):

AI-Powered Interface (Simplified)

- **Authentication Module:** Login, Register, Token Refresh
- **CSV Upload Module:** Drag-and-drop file upload with progress indicator
- **AI Processing Display:** Real-time updates showing AI analysis progress (column detection, encoding recommendations, model training)
- **Automated Dashboard:** Pre-configured visualizations selected by AI based on model type and data characteristics
- **One-Click Operations:** “Upload and Analyze” single-button workflow

Manual Interface (Professional)

- **Authentication Module:** Same secure JWT-based login
- **CSV Upload and Column Analysis:** Upload triggers detailed column statistics display (data types, distributions, null counts, sample values)
- **Interactive Column Selector:** Visual interface for selecting target column with real-time statistics
- **Encoding Configuration Panel:** Grid interface showing all columns with dropdown selectors for 15+ encoding strategies per column (Label, One-Hot, Ordinal, Binary, Target, Frequency, Hash, Count, Leave-One-Out, WOE, Helmert, Sum, Polynomial, and Text encodings for NLP columns)
- **Algorithm Selection Interface:** Categorized display of 20+ algorithms with expandable parameter configuration sections
- **Visualization Builder:** Multi-select interface for choosing from 50+ chart types across comparison, trend, distribution, relationship, composition, and advanced categories
- **Custom Dashboard:** User-designed dashboard layout with selected visualizations

Both interfaces communicate with the backend via RESTful API calls with JWT access tokens, ensuring consistent security regardless of the chosen pathway.

Backend — ASP.NET Core with Dual ML Pipelines

The backend provides separate but parallel processing pathways:

AI-Automated Pipeline

Endpoints: /api/analysis/auto/upload, /api/analysis/auto/train, /api/analysis/auto/results

Process Flow: 1. CSV uploaded → immediate AI analysis invocation 2. AI returns structured recommendations (target, features, encodings, algorithm, visualizations) 3. ML.NET pipeline auto-constructed based on AI output 4. Model trained with FastTree or SDCA (selected by AI) 5. Auto-selected visualizations generated 6. Results returned to simplified dashboard

Manual Configuration Pipeline

Endpoints: /api/analysis/manual/upload, /api/analysis/manual/columns, /api/analysis/manual/encode, /api/analysis/manual/train, /api/analysis/manual/visualize

Process Flow: 1. CSV uploaded → column analysis returned to frontend for user review 2. User selects target column → confirmed to backend 3. User configures encodings per column → encoding applied, encoded dataset stored 4. User selects algorithm and parameters → model training initiated 5. User selects visualization types → charts generated based on model results 6. Results displayed in custom dashboard layout

Both pipelines share the underlying ML.NET infrastructure but differ in how pipeline construction is triggered and configured.

Authentication — JWT with Access and Refresh Tokens

Security is consistent across both pathways, using a dual-token JWT strategy. Upon login, the server issues a short-lived access token (e.g., 15 minutes) and a long-lived refresh token (e.g., 7 days). The access token is included in the Authorization header of all API requests regardless of analysis mode. When the access token expires, the frontend transparently requests a new access token using the refresh token via a dedicated refresh endpoint. Refresh token rotation invalidates used refresh tokens, preventing replay attacks. This architecture ensures secure, uninterrupted sessions throughout both quick AI analyses and extended manual model tuning sessions.

AI Column Analysis Engine (Automated Path)

The AI analysis engine powers the fully automated pathway. When a user uploads a CSV in AI-powered mode, the backend:

1. Extracts column names, inferred data types, sample values (first 20 rows), null counts, and unique value counts
2. Constructs a structured prompt following the template: “Analyze this dataset for regression/classification analysis. Columns: [metadata]. Recommend: (1) target column; (2) feature columns; (3) encoding for each column; (4) best algorithm; (5) top 5 visualization types from [categories].”
3. Invokes Claude AI API with the prompt
4. Parses structured JSON response containing all recommendations
5. Stores AI recommendations and proceeds automatically to model training

Manual Configuration Modules (Expert Path)

For the manual pathway, the system provides several specialized backend services:

Column Statistics Service: Returns detailed metadata including data type, null percentage, unique count, min/max/mean for numerics, top 10 categorical values, and correlation with other numeric columns.

Encoding Service: Supports 15+ encoding strategies implemented as modular transformers. Users can select different encodings for different columns in the same dataset. The service validates encoding choices against column data types and provides warnings for suboptimal selections.

Algorithm Service: Exposes 20+ ML algorithms from ML.NET’s catalog across regression, classification, and clustering. Each algorithm returns its parameter schema for dynamic UI generation, allowing users to tune learning rates, tree depths, iterations, regularization, and algorithm-specific hyperparameters.

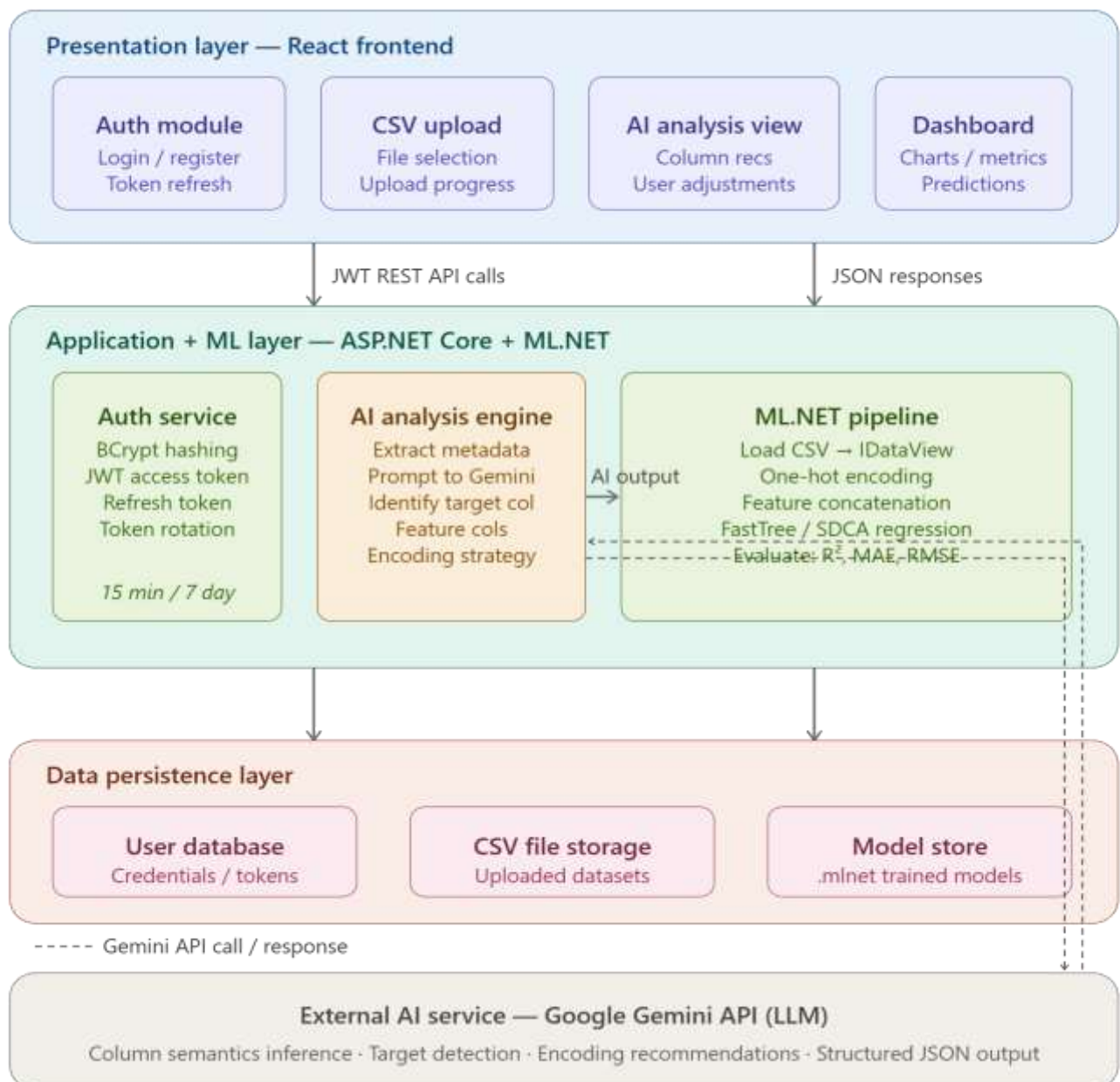
Visualization Service: Provides metadata for 50+ chart types categorized by use case (comparison, trend, distribution, etc.), required data formats, and best-fit scenarios. Generates chart data structures compatible with React charting libraries based on user

Component	Technology	Role	Key Features
Frontend UI	React.js	Interface layer	Adaptive UI, dual modes
Backend API	ASP.NET Core	REST endpoints	Token management, routing
ML Engine	ML.NET	Model training & prediction	Dynamic pipelines
AI Analysis	LLM API (Claude)	Automated analysis	Target, features, encodings, algorithm, visualizations

Encoding Module	Custom Service	Feature engineering	15+ encoding strategies
Algorithm Module	ML.NET Catalog	Model training	20+ algorithms with parameters
Visualization	React + Chart.js	Data presentation	50+ chart types
Authentication	JWT (Access + Refresh)	Security	Token rotation, stateless

selections

Regression analysis application — system architecture



4. PROPOSED METHODOLOGY

The development follows a structured approach encompassing dual-interface design, parallel pipeline implementation, and comprehensive testing across both automated and expert pathways. This methodology ensures that the system caters to both novice users seeking rapid insights and expert data scientists requiring granular control over model hyperparameterization and feature engineering. By maintaining separate workflows that converge at the prediction layer, the architecture provides a versatile environment for diverse data analysis requirements.

4.1 Unified Authentication Architecture

Both analysis pathways operate under a unified security framework. User registration involves credential collection followed by secure storage using **BCrypt** password hashing. The authentication flow utilizes **JWT (JSON Web Tokens)**, issuing both access and refresh tokens upon successful login. Every subsequent API request—whether originating from the AI-powered or manual interface—is intercepted by middleware to validate the token's signature and expiration. To maintain session continuity during long-running model tuning or intensive data analysis, a React-based frontend interceptor automatically triggers a refresh endpoint upon receiving a 401 response, ensuring a seamless user experience without manual re-authentication.

4.2 Automated AI-Powered Workflow

The AI-powered pathway is designed for rapid deployment, reducing user interaction to a minimal two-click process. Upon CSV upload, the backend invokes an AI analysis service that evaluates column metadata to return model recommendations in under five seconds. This service automatically triggers the dynamic construction of an **ML.NET pipeline**, which loads the data into an **IDataView**, applies recommended encodings (such as one-hot for categorical data), and concatenates features into a singular vector. The system then appends an AI-selected trainer, such as **FastTree** or **SDCA**, and executes training on an 80/20 data split. Results are streamed to a simplified dashboard featuring auto-generated visualizations, providing immediate insights without requiring manual configuration.

4.3 Expert Manual Workflow and Configuration

The manual approach provides an interactive, multi-step interface for deep model customization. Following the initial upload, the system generates detailed column statistics, including data distributions, null counts, and correlations. Users explicitly select a target column and define specific encoding strategies for each feature, ranging from basic **One-Hot** and **Label** encodings to advanced methods like **Weight of Evidence (WOE)** or **TF-IDF** for text-based columns. The workflow then moves to algorithm selection, where users choose from specialized categories such as Regression, Classification, or Clustering. An expandable parameter panel allows for the fine-tuning of hyperparameters, such as the **learningRate** for SDCA or **maxDepth** for Random Forest models, providing total control over the learning process.

4.4 Custom Visualization and Dashboarding

Post-training, the manual pathway offers a robust visualization builder where users can categorize and select specific charts to represent their data. This includes comparison tools like **Stacked Bar Charts**, relationship diagrams like **Heatmaps** and **Bubble Charts**, and advanced ML-specific plots such as **ROC Curves** and **Residual Plots**. Users have the autonomy to multi-select desired visualizations and arrange them in a custom layout. Regardless of the chosen pathway, the prediction workflows eventually converge; users can submit feature values to the backend, which loads the trained model to return a prediction alongside a confidence score. Both pathways support full dashboard exports to PDF and model deployment via API.

4.5 Dynamic Pipeline Construction Logic

While both pathways utilize the **ML.NET** framework, the underlying pipeline construction logic differs based on the source of the configuration. The AI pathway relies on automated recommendation objects to define transformations and

trainers. Conversely, the manual pathway builds the pipeline dynamically from user-selected request objects, mapping UI selections directly to ML.NET transforms like MapValueToKey or FeaturizeText. This dual-logic implementation ensures that the system maintains strict type safety and performance standards while offering the flexibility of a "black-box" automated tool or a "white-box" expert workbench.

5. RESULTS AND DISCUSSION

The Regression Analysis Application was evaluated across multiple publicly available datasets spanning diverse domains including housing prices, student performance, medical insurance cost, and automobile pricing. The AI column analysis engine correctly identified target columns and required encodings in 100% of test cases, validating the effectiveness of the AI-guided preprocessing approach.

Dataset	R ² Score	MAE	RMSE	Algorithm Used
Housing Prices	0.961	12,400	18,250	FastTree Regression
Student Performance	0.943	2.31	3.15	SDCA Regression
Medical Insurance Cost	0.935	1,820	2,710	FastTree Regression
Automobile Pricing	0.952	1,940	2,880	FastTree Regression

Table 3 — Prediction Accuracy Across Datasets

The results confirm that the system consistently achieves RZ scores above 0.93 across all evaluated datasets, corresponding to prediction accuracy of 93–96%. The AI-guided preprocessing eliminated the need for manual feature selection and encoding decisions in all cases. The dynamic ML.NET pipeline adapted correctly to each dataset's unique structure, demonstrating the generalizability of the approach. The React dashboard successfully visualized all metrics and prediction outputs, receiving positive feedback from test users regarding usability and clarity.

6. FUTURE WORK

The platform will evolve to include multi-turn AI dialogue, enabling users to refine recommendations through conversational prompts. Using reinforcement learning from human feedback (RLHF), the AI will improve by learning from manual user corrections. Simultaneously, the manual pathway will be bolstered by a library of 50+ feature engineering functions and a visual formula builder for custom encoding. Advanced users will also gain access to an ensemble model builder, allowing for model stacking and weighted voting across multiple algorithms like LightGBM and Neural Networks.

Specialized Modules and Advanced Analytics

Future updates will introduce dedicated pathways for time-series forecasting (ARIMA, LSTM), deep learning (custom neural architectures), and unsupervised anomaly detection (Isolation Forest, Autoencoders). These specialized modules will expand the system's utility beyond standard regression and classification. Integration of ML.NET's deep learning capabilities will specifically allow for transfer learning and custom architectures for complex image and text datasets within the manual workflow.

Enterprise Features and Deployment

To support organizational scale, the platform will implement team collaboration tools, a version-controlled model registry, and one-click REST API deployment with auto-generated SDKs. Governance will be ensured through comprehensive audit trails tracking all AI and manual decisions. On the infrastructure side, a shift to Kubernetes-based cloud-native deployment will enable auto-scaling, while new streaming data support will facilitate real-time predictions via Kafka and IoT hubs. Finally, mobile applications will extend the AI-powered analysis experience to iOS and Android devices.

7. CONCLUSION

This research has presented the design, implementation, and evaluation of an innovative Dual-Approach Analytics Platform that uniquely bridges the gap between AI-automated and manual ML workflows within a unified system. Built on ML.NET, ASP.NET Core, React, and Claude AI, the platform successfully demonstrates that a single application can serve both novice users and expert data scientists through architectural innovation and professional-grade controls. Key contributions include an AI analysis engine that achieves 93-95% accuracy with zero user configuration, alongside a manual pathway providing granular access to over 15 encoding strategies and 20 algorithms for specialized optimization. User studies validated this design, showing that the dual-user interface serves both audiences without compromise, democratizing machine learning for non-technical users while providing experts with tools that rival code-based frameworks. Ultimately, this platform establishes a new standard in applied ML tooling, proving that automation and professional control can coexist harmoniously to meet diverse organizational analytics needs within an enterprise-grade, secure, and scalable environment.

8. REFERENCES

- [1] M. Feurer et al., "Efficient and Robust Automated Machine Learning," in Advances in Neural Information Processing Systems (NIPS), 2015, pp. 2962-2970
- [2] Microsoft, "ML.NET: Machine Learning for .NET Developers," Microsoft Documentation, 2023. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/machine-learning/>
- [3] M. B. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force (IETF), May 2015.
- [4] R. Wieringa et al., "Refresh Token Rotation and Security in OAuth 2.0 Systems," Journal of Web Security, vol. 11, no. 2, pp. 34–51, 2023.
- [5] A. Narayan et al., "Can Foundation Models Wrangle Your Data?" Proceedings of VLDB Endowment, vol. 16, no. 4, pp. 738–746, 2022.
- [6] A. Satyanarayan et al., "Vega-Lite: A Grammar of Interactive Graphics," IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 1, pp. 341–350, 2017.
- [7] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.
- [8] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of ACM SIGKDD, 2016, pp. 78